Global Information Assurance Certification Paper

Humans Are Always The Weakest Link

Kevin Swaim
July 8, 2004
GSEC Practical Requirements (v.1.4b) (August 2002) Option 1

ABSTRACT

Technological means of protecting your network infrastructure are a good start, but by themselves are inadequate. There are so many ways to defeat network security by exploiting people that a term was even coined for it--*social engineering*. A good social engineer can use their people skills to attack you because humans are inherently unpredictable. Humans are always going to be the weakest link in your security model.

In the following two parts, I will try and show how a combination of social engineering and user mistakes can lead to your infrastructure being exposed to risk. Part one will deal with physical security, and part two will deal with user error.

Part One.

Physical security receives lots of attention, as well it should, and with the use of sophisticated devices it is assumed that the infrastructure should be secure. Can that always be said to be true?

Mantraps.

A mantrap is a device that allows a single authorized person access to a secured facility. For this paper, we will be discussing the revolving door mantrap, keyed by a proximity badge. Mantraps in this fashion are often used as the first line of defense into the company's physical plant, data center, or network operations center (NOC.) Often, the mantrap is used in conjunction with a booth for security guards. At the very least, a camera is usually deployed and the door is observed by security guards within the facility.

The mantrap works by allowing a single person into one of the revolving chambers of the door. Access is granted by a proximity badge. As the door revolves, it discharges the single occupant on the other side. This person now has access to the interior of the building. If a person does not have the correct access, the door will not revolve and allow admittance. It also works to control access out of the building as well, and, if the person does not have access, will not allow access out. This system usually has a bypass for deliveries, people carrying large items, handicapped access, people who have lost their proximity badge or for whom the proximity badge has become inoperable, or for emergency events like a fire or storm event.

An easy way to circumvent this device is to have one person activate the door with their proximity badge, and for that person, plus one other, to enter the chamber. This will discharge both occupants into the building, with free reign once inside. When two people enter the chamber, the door should reverse direction and lock down, causing the occupant to be deposited back outside the building and forced to enter through the bypass. It is possible, however, for two people, quick footed and agile, to make their way through the door without exerting enough force to trigger the door to reverse direction. I have seen this done several times, and only once did the door reverse direction.

The human element, the security guards, watched as this event played out, and did not respond. This is the real weakness. After a brief conversation with the guards, I determined that they did not understand the full purpose of the mantraps, and their place in the security process. The guards should be trained in the mantrap's purpose, and should stop two people from entering the same chamber. They did not. They should have noted the identity of the two people in the mantrap, made their managers aware of their attempt to defeat the mantrap, and consequences should follow. This did not happen. The guards assumed that the mantraps work correctly if they allow people in or out. The guards also

assume that as long as people are allowed in or out, that the people must have the correct access, have been authenticated correctly, and that the process of physical access to the building is working correctly.

The guards should act as another tier of security. If they are not adequately trained in the physical process, they do not act as a second tier. Instead, the mantrap is providing a single tier of security to physically access the building.

It is possible to purchase a mantrap that weighs the occupants, making sure that a sole occupant is allowed in and out of the mantrap. Another option is a mantrap that uses infrared to detect unauthorized entrants in the mantrap. Once such product, the Tourlock, is sold by Boon Edam.[1] With such a system in place, it is harder to achieve unauthorized access through the mantrap itself, but there is another way to gain access that doesn't involve the mantrap at all.

Usually, if there is a mantrap, there is a bypass method for emergencies, deliveries, or people carrying items that won't fit in the mantrap. A method I have used during penetration testing is to carry a large box or package, and walk up to the bypass. The security guards who man the bypass gave a cursory glance and unlocked the door. My badge was obscured by the box I was carrying. When I questioned the security guard later, he said he was simply trying to help out.

Proximity Badges

For this paper, we will be discussing the proximity badge as used by many companies. This is a single form of authentication, and works by a the badge reader using RF energy to power up the chip in the card and conveying back a number that is compared to a database to allow access.[2]

A typical technological hack might be to purchase a badge reader over the Internet. I found that by going to Google, and typing in Proximity Badge Reader, the first hit took me to a company called RFIDeas, Inc. that sold badge readers.[3] These readers work both on HID and Motorola formats. With such a reader, you would acquire a badge, read it, decode the scheme for figuring access, then reprogram the badge to give you the access you need.

This is a lot of work though, and there is a much simpler way to achieve access to the secured area. A simpler method, and one that I have utilized and observed in penetration tests, is to ask someone else to use their badge to provide access to the building. In one case, I had a badge that belonged to a different company. I tried badging in several times, then asked the next two people if they could badge in for me. The second person replied with, "I hate when that happens," then used their proximity badge to open the door for me.

[1] Boon Edam.
[2] RSA Security.
[3] RF IDeas, Inc.

Security guards, who were watching the whole thing, did nothing about this. This time there was a two-fold failure, one by the employee using their badge to allow me in, and the second by the security guards who did nothing about it.

Another way that human error can lead to a compromise with a proximity badge is if the badge has a photograph applied to the badge. As an example, during a renewal of the photo on my proximity badge, I asked for a second copy of the photo. These photos are printed from a PC on to a plastic laminate that is then fastened to the proximity badge. I found a person who had access to a room that I could not access, then waited until the person took his badge off and set it on his desk. Why he did this is unknown. Perhaps the badge was in his way, or bothering him, but for whatever reason he left it on his desk and walked away. I "borrowed" his badge, and stuck my self-adhesive photo on his badge, then badged in to the room that I could not access using his proximity badge but with my picture. Once in the room, I was questioned by someone who knew I did not have access to the room. I walked out, then badged back in, which seemed to answer any question the person had about whether I had the correct access and whether I belonged in the room. When asked later, the person responded, "Well, your badge got you in to the room, and your picture on the badge matched, so I figured it was okay."

These were both easy exploits of the system, and were made possible by several human mistakes. First, no one should ever use their access card to allow someone else access to the building. People must be trained to inform the security guards if someone ever asks them to do such a thing. The person who can not badge in should be forced to go to the bypass where a trained security guard should act as a second tier of security. Second, no one should ever take their badge off and leave it unattended. Third, if a person notices someone in an area and does not recognize them or knows that the person does not have access to the room, they should escort them to security, where the guards can check the badge and make sure that the badge has not been tampered with and that the access on the badge matches the access the person has for the building.

A better solution might be to purchase a smart proximity badge. These badges can allow the user to authenticate using information like passwords, PIN's, and biometrics. This is known as two factor or three factor authentication.[4] This would have made the second exploit impossible, because I would not have been able to guess the password, PIN, or meet the biometric information of the user. However, the first exploit would still be possible. User training and properly trained and vigilant security guards should act as a second tier of security.

Publicly Accessible Computers.

[4] RSA Security p. 7.

Many companies have computers or terminals accessible to the public. This is case where a company's desire to more closely communicate with their customers can lead to an easy exploit. For this example, I will use a financial company that asked me to perform a security audit.

The company had a communal reception area, shared by 3 other companies, each with a reception desk and a receptionist. This is a common setup in industrial parks, as many companies can't afford rent on their own office building. The reception area had a public entrance, and behind each reception desk there was an entrance to the respective company. I entered the reception area through the public entrance instead of entering from one of the private entrances behind the reception desk. Three of the reception desks were empty, including my target who I will call Company A. The fourth desk, another financial company, was occupied. The test went like this:

"Hello, I'm Kevin Swaim. I work for Company A. Where is Receptionist A?" Receptionist A was the name on the reception desk of Company A.

"Hello, I'm Receptionist B. I'm with Company B. They," she waved around, "went out to lunch."

"They didn't invite you?"

"Well, I'm new here. This is my first week."

"Still, that's not very nice. You would think they could have invited you along. I know I would have. You always should make the new people feel welcome. Receptionist A can be kind of unfriendly to the new people. It's best just to ignore her. After a couple of weeks of that, she will open up."

Receptionist B smiled. "Yeah, she hasn't really said much to me. They have their own little lunch group here, and they go out every day at noon, but nobody invites me. I'll just wait them out."

This is a textbook case of social engineering.[5] By noticing the other receptionists' lack of invitation to lunch, and by a lucky guess that reaffirmed her opinion of the others, I had established a trust. This would make other information easy to acquire.

I figured they took an hour lunch, as there were no restaurants nearby, which meant I had very little time. "Well, I need to fix her machine. She hasn't complained has she?"

"Not that I've heard."

[5] Granger, Sarah.

I sat behind the reception desk for Company A. I quickly tried a few passwords, like "password" and the user's name, but none worked. Afraid they might have an account lockout set, I quit trying passwords.

"Well, I don't know her password," I said to Receptionist B, "and I can't log her off. She might be in the middle of something, like a memo."

"She usually types things for the president's secretary, so that could be bad."

"Okay, I'll try again later. I have a ton of work to do. Maybe I'll put it off until tomorrow." Before I left, I installed a keystroke logger. A keystroke logger is either software or hardware that records keystrokes on the computer[6]. I installed a hardware keystroke logger, as the cabling on the computer was exposed in the back. A quick search on Google will lead you to sites like www.keyghost.com, where you can purchase an equivalent type of keystroke logger.

Taped to the side of the monitor was a list of all the employees in the company, their responsibilities, and phone number. It seems that the receptionist also worked answering the phone. One of the entries was for a woman named Secretary A, the president's secretary. "See you later."

I left, once again through the public entrance. Later in the day, I called back and Receptionist A answered. I claimed to be from CDW, and asked who was responsible for paying bills. The answer was Secretary A, the president's secretary. I hung up while being transferred.

I came back the next day at 20 minutes past noon, entering through the public entrance. Once again, Receptionist B was there by herself. "Hello, how are you? They left you by yourself again, huh?"

"Yes. Back to fix her computer?"

"Yes." I sat down and checked the keystroke logger. I found her password, logged on as her and went to her network drives. There were several. "Hey, Receptionist B, do you know what drive Receptionist A is using for documents she creates? I could go back and check, but it would take me a few minutes."

"I heard her telling another girl she has an M: drive."

I checked her M: drive, and found several folders, one called Secretary A. I checked under the folder Secretary A and found Word and Excel files. I found a text file call passwords.txt that contained the login names and passwords of all the executives. I dumped this file and all the others to a USB drive on my keychain, unplugged the keystroke logger, then told Receptionist B the problem was fixed. I left through the public entrance.

[6] Wang, Wallace.

When I got back to my office, I found Word documents that contained enough information to log on remotely to their Windows domain, and also to their Unix mainframe, both with administrator privileges.

The social engineering used was just part of the attack. Company A had made several mistakes: they had a machine that sat in a publicly accessible place, the cabling was accessible to the public, and there was a listing of all the names and phone numbers taped to the side of the monitor. The final mistake was the lack of human oversight. Not only was the machine left unattended by an employee of Company A, but the person who was left behind, a person not affiliated with the company, did nothing to stop the exploit. Instead, the receptionist from Company B helped with information that was used to exploit Company A. When showed the report, Company A's CIO replied, "The workstation asked for a password didn't it? I thought that meant we were safe."

Suggestions

A security policy is a must for any company, and must take into account both logical access to the network and physical access to the building. It does no good to have heavily protected servers and workstations, but allow anyone physical access to the data center that houses them. Once a security policy is written, then testing and auditing must be done. A penetration test is a great tool that will show any weakness in your security policy.

A good penetration test will show both gaps in the logical access to your network and the physical access to the building and machines, and also how to fix those gaps. Once a penetration test is complete, it is necessary to implement changes that will protect you from the findings of the penetration test. You must close the gaps. A penetration test that does not change the way you handle security does not help anyone.

As an example, I pointed out several gaps in physical security at a company. After several months, I talked with an executive at the company. None of the gaps were closed, none of the suggestions implemented. I inquired as to why. The executive responded, "We thought it would take too much time to implement the changes. We also don't have the money." I pointed out that the gaps still existed, and offered to work with the company to find cheaper and faster alternatives that would still offer them adequate protection. The executive declined.

Many of the previously mentioned exploits could have been solved with minimal time and cost. A brief training session with the guards could prevent many of the physical access problems to the building. A locked down computer and a policy to never leave it unguarded would have prevented the access to the company's systems. In each case, the human element, which should serve as a last chance

to prevent an intrusion, did not function. This is where time and effort must be spent. It is too easy for a person to see a mantrap and think that it is performing its function. It is too easy to see a proximity badge and think that they are infallible. It is too easy to assume that because a workstation is locked and requires a password that the workstation is safe. Until the, humans will always be your weakest link.

Part Two

From the user who forwards on a virus just so the user's friends will know what not to open, to the user who mistakenly brings in a virus on his laptop, user error is the biggest loss of productivity due to security failures.

Patches

In his paper, "10 IT Nightmares & How to Avoid Them," Paul Chard lists product vulnerabilities and patch management as number six[7]. With the variety of exploits and virii released each year, it would seem foolish not to update or patch your software. Most environments have some form of patch management in place, but it is often thwarted by the end user.

One user I called had a problem with his laptop. Every time it would start, it would attempt to install some software on his machine. Every time the user would shut down the software, as he was afraid someone might be installing something that was not authorized. In this case, it was a patch to protect him from the vulnerability listed in Microsoft's security bulletin Security Bulletin MS04-011[8]. It came as no surprise that the laptop had been infected by Sasser, a virus that exploits the LSAS vulnerability listed in this security bulletin.[9]

This is an example of how user behavior could not be predicted. It was assumed by all involved in the patch process that the patch would be deployed automatically to the users, and that if the patch started to install, then the patch would be allowed to complete and the patch would be successfully installed. No one counted on the end user stopping the install. When explaining how important the patch was, the user replied with this, "That's why you installed this AV software, right? Why won't it do the job? Do you really need the patch?"

Another example of this came when I performed a security audit on a company and found that they were not patched for several Microsoft vulnerabilities. This seemed odd, as the company had a technician who performed patch management for the company. After glancing through his patch management of

[7] Chard, Paul.
[8] Microsoft.
[9] Shannon, Heather.

the past several years, there was disturbing trend--there were several gaps in patches. After talking with the technician, I found they only applied one patch a year for Microsoft products, as the technician thought the patches were cumulative. When I explained that the Service Packs were cumulative but the patches had to be applied as they were released, he explained that they had several virus outbreaks for vulnerabilities he assumed they had patched. He had been at a loss to explain how the virus outbreaks were occurring.

This is a problem that has recently become apparent. Users expect one piece of software to fix their problems. People fail to realize that patching software is an ongoing process. Patches are like virus definitions. It requires a large company's full attention to make sure everything is kept up to date.

Virii and AV Software

Many, if not most, security professionals have worked a virus incident. The pattern is almost always the same. User calls and complains, technician checks and finds virus active on the system. Another common occurrence is to call the user and hear comments like, "Well, I knew I shouldn't open that attachment, but I did it anyway." With the rise of virii that spoof email addresses, such as W32.Sober, W32.Netsky, and MyDoom, you now hear things like, "I didn't think my friend would send me a virus. My AV software said it was a virus, but I thought the AV software was wrong."

Users are prone to opening attachments they should not, but this risk can be mitigated by blocking all attachments that are unnecessary at a company's firewalls. Also, in an enterprise environment, a rapid system for deploying virus definition files is a must. However, in most AV software, if the software identifies the virus, users are still allowed the option to open it. Because users are likely to trust people they know or names they recognize, they will open files and mistakenly think that the AV software is wrong.

Of course, you could have users who turn of their AV software. Many users have complained to me that the AV software slows down their machine, the AV software uses up their system resources, or that they just don't trust the AV software. One user said he thought the AV software might be monitoring his web surfing.

A well-written virus that bypasses most AV software is no longer necessary to exploit a system, just a random user who turns off the AV software and receives a copy of the virus in their email is all that it takes to exploit your system, especially with the newer multi-exploit virii that takes advantage of an unpatched machine in your network.

Other User Mistakes

There are several other mistakes that users make. One example Kevin Mitnick refers to as the reverse sting, which he defines as "A con in which the person being attacked asks the attacker for help."[10] A very primitive version of this seems to be exactly what happens when you receive spam with an unsubscribe option at the bottom. When you click on the unsubscribe button, you are asking for help. You are asking the author of the email to take you off the mailing list. However, this does not help you, but only the spammer, as you will almost immediately start receiving a large amount of spam. You, the recipient of spam, are validating your email address to the spammer. You are, in effect, helping him do his job. Since spam is often sent through spam relays which are prone to infection by virii, you can rest assured that much of your incoming spam will now be infected.

Users will often open email from an address they don't recognize. This is quite common, as humans want to help and please others, and most users think it bad manners to ignore an email. They do not want to miss an email that is legitimately addressed to them. However, a new technique spammers are employing puts a single-pixel jpeg image that is hyperlinked back to their server. Since the jpeg is a single pixel, you probably would not notice it, and when you open the email it connects to their server and validates your email address. There are several technical ways to combat this, from requiring all outgoing connections to force a logon to your proxy server, to changing your email to only open in text format. However, an easier solution is to tell your users not to open email from addresses that are unfamiliar to them.

Of course, internal email addresses should never by allowed outside of a company unless absolutely necessary. Spammers can try and make up addresses, but it is easier to harvest them off web sites or newsgroups. I know of several people who have used their internal email address to post a message on a forum or newsgroup, and suddenly their inbox is flooded with virus-infected spam.

Internal email addresses posted in forums or newsgroups related to technical matters is also an easy exploit. Try surfing through forums related to routers or intrusion detection systems, and see how many company addresses you will see. Why bother finding a technical way to hack in to a company, when you can search through a forum on Cisco routers and see that a technician from Company X is asking how to block access on certain ports of the company's external-facing routers. Now you have the open ports and the company's names, and, using the reverse sting again, send the person asking the question malware in the guise of a patch. When they run it, it will open a backdoor for you to exploit.

A company I know was recently infected with a virus that was brought in from the outside via a user's laptop. The user took the laptop home, plugged in to their DSL line, became infected, then when plugging back in to their company's LAN,

[10] Mitnik, Kevin. p. 133.

proceeded to infect thousands of machines. This totally bypassed the company's perimeter defenses, as the company had never envisioned an attack from occurring from within the company. Once again, someone did something totally unexpected. At the bare minimum, a personal firewall should be running on the user's machine. If the company can afford it, an investment in a hardware firewall for home is not out of line.

Solution

The acceptable use policy is a must to help prevent user mistakes from compromising your system. There are examples of acceptable use policies on the web. A good beginning can be found at SANS.[11] This is the bare minimum that a company should put in place to protect itself.

Every company should also be prepared to punish those who violate the acceptable use policy. Through regular auditing and penetration testing, the company should make sure that the acceptable use policy is not being broken. It does no good to have an acceptable use policy if anyone can violate it without any repercussions. According to CERT, "Gain management-level support for the development and promulgation of an acceptable use policy."[12] This means that it is not enough just to have the policy, but the policy must have teeth. If management does not buy in to the importance of the security policy, then it does no good to inform users of forbidden behavior. Until you have trained your users and provided them with an acceptable use policy, humans will always be your weakest link.

Summary

There are untold ways to compromise a company's security, because there are untold ways for people to do things that you simply can not plan for.

There is something you can do to protect yourself--train the people in your company to recognize threats and to take them seriously. Humans can be your weakest link, but with the right training, they can also act as a last safety net. People rely on technology as a crutch. From the security guard to the end user, people make mistakes because they assume the technology is going to make them safe. People must be made aware that they play an important part in the security of the company. You do not need a company of heroes to make a difference. All you need to make a difference is educated, diligent employees who seek to protect the security and assets of a company. With the proper training and planning, you can keep the humans in your organization from being the weakest link.

[11] SANS.
[12] CERT.

References:

Boon Edam, "Tourlock 180" URL:http://www.boonedam.nl/inc/index.htm (July 9 2004).

Cert "Develop and promulgate an acceptable use policy for workstations" 1999 URL:http://www.cert.org/security-improvement/practices/p034.html (July 13 2004).

Chard, Paul. "10 IT Nightmares & How to Avoid Them." June 13 2004. URL:http://www.itsecurity.com/papers/avanade1.htm (July 11 2004).

Microsoft. "Microsoft Security Bulletin MS04-011." V2. April 13 2004. URL:http://www.microsoft.com/technet/security/bulletin/MS04-011.mspx (June 12 2004).

Mitnick, Kevin. The Art Of Deception. Indianapolis: Wiley Publishing, Inc. 2002. p. 133.

RF IDeas, Inc. URL:http://www.rfideasstore.com/pcproxproxre.html (July 9 2004).

RSA Security 2002 "RSA Smart Badging: Securing PC's, Networks, and Buildings" URL:http://www.rsasecurity.com/products/keon/whitepapers/SDSB_WP_1102.pdf (July 9 2004).

SANS InfoSec Acceptable Use Policy URL:http://www.sans.org/resources/policies/Acceptable_Use_Policy.pdf (July 14 2004).

Security Focus "Social Engineering Fundamentals, Part I: Hacker Tactics." Sarah Granger December 18 2001 URL:http://www.securityfocus.com/infocus/1527 (July 10 2004).

Shanon, Heather .Sasser.B.Worm May1 2004 URL:http://securityresponse.symantec.com/avcenter/venc/data/w32.sasser.b.worm.html (June 14 2004).

Wang, Wallace. Steal This Computer Book 3. Berkley: Publishers Group West, 2003, p112-113.